FERMILAB-Pub-90/192

# Primary Vertex Finding in Proton-Antiproton Events with a Neural Network Simulation *

Clark S. Lindsey and Bruce Denby
*Fermi National Accelerator Laboratory*
*P.O. Box 500*
*Batavia, Illinois 60510*

September 1990

* Submitted to Nucl. Instrum. Methods A.

# Primary Vertex Finding in proton-antiproton Events with a Neural Network Simulation

Clark S. Lindsey, Bruce Denby

*Fermilab*, Batavia, Il. 60510, USA*

**Abstract:**

In a test of neural networks in high energy physics pattern recognition problems, drift chamber data from experiment E735 at the Tevatron proton-antiproton collider was used in a neural network simulation to find the primary event vertex. A three layer, feed-forward neural network was trained with the back-propagation technique to give the beam line vertices of tracks traversing sub-sections of the chamber. Summing the outputs of all sub-section networks then gives the primary vertex along the beam line. Results are compared to conventional methods.

## 1. Introduction:

The use of neural networks to perform pattern recognition has become an area of intense study. Neural networks have been applied, for example, to recognition of handwritten characters [1] and sonar returns [2]. An advantage of neural networks over most conventional techniques is their inherent parallelism, which can provide very fast performance if executed in neural network hardware. Also, some neural network architectures, such as the feed-forward network trained by the back-propagation technique, learn from examples. Such networks are flexible and adaptible and can perhaps perform tasks where no conventional algorithm is known.

Recently, results of tests of neural networks in high energy physics pattern recognition problems have been presented. Examples include track finding in drift chambers [3], electron identification in calorimeters [4], and determining the slope and vertex of tracks [5]. Although conventional approaches to these problems are generally satisfactory, neural networks could have advantages where fast performance is required such as in on-line trigger applications and as off-line analysis "accelerators" to help process the very large data sets produced in many modern high energy experiments.

The neural network tests so far, including the work presented here, have been simulations on sequential computers and so offer no speed advantage yet. However, neural network VLSI hardware is now becoming available and will provide the full parallel performance of the network architectures.

One difficulty in applying neural networks to high energy physics pattern recognition problems is that there is often a great number (hundreds to thousands) of signals from the detector systems. The hardware neural networks will be limited to less than a few hundred neurons for the near future. Also, if the training is done first by simulations on sequential machines and the interconnection strengths then loaded into a hardware network, the training technique takes increasingly long to run with increasing numbers of neurons and interconnections. However, for some tasks it is possible to divide the detector into identical sub-sections. A neural network is then trained on only the sub-section, and summing the outputs of all sub-section networks gives the final answer.

We test this approach here by determining the primary vertex along the beam line of proton-antiproton events using information from a drift chamber located next to the beam pipe. The transverse position of the vertex is restricted to the beam size (about 1mm radius) so only the one dimension (z) along the beam line needs to be determined. The vertex can occur over roughly a 1.5m range. The chamber is divided into overlapping sub-sections. Using the drift distances from the wires, the net is trained to determine the intercept along the beam line of tracks that traverse a given sub-section. The overlapping sub-sections are then processed through the network and the sum of their outputs gives a vertex distribution along the beam line. The position of the maximum of this distribution is taken as the primary vertex. (Here we take "primary" vertex to be the position along the beam line of the proton-antiproton collision and "track" vertex to be a given track's intercept on the beam line which may or may not coincide with the primary vertex.)

Fast determination of the primary vertex has applications in triggering. For example, one might want to restrict the primary vertex to a narrow section of the beam line for optimum detector acceptance. Also, searches for fast secondary decays might need to have first the primary vertex position. Another possible application could be at the SSC where there will usually be multiple interactions per beam crossing. The vertex distribution mentioned above would have peaks at each primary event vertex.

## 2. Vertex Chamber in Experiment E735:

A schematic cross-sectional view of the E735 detectors in the C0 interaction hall at the Tevatron collider is shown in fig. 1a. The detector system has been described in detail elsewhere [6,7]. It consists basically of two parts: (1) central detectors surrounding the interaction point for determining the charged multiplicity; and (2) a spectrometer to the side for momentum and mass-identification analysis of charged particles emitted in the central region. A planar drift chamber called the "z-chamber" on the spectrometer side of the beam pipe was used to find the primary vertex and also to help track particles entering the spectrometer [8].

Fig. 1b shows a plan view of the detectors. Besides the z-chamber, primary vertex information is also available from tracks in the spectrometer and from the time of flight (TOF) scintillator counter arrays located around the beam pipe on each side of the the interaction region. The times of flight of charged particles from the interaction point to the counters are used to determine the interaction time and primary vertex.

The z-chamber is 1m in length along the beam pipe. There are three layers of wires with the layers separated by 1.1cm (fig. 2a). The innermost layer is 13.1cm from the beam line. The wires are 10cm long, running vertically in fig.1a and normal to be beam line. There are 96 sense wires per layer and spaced 1.1cm apart. A single field shaping wire (not shown in fig. 2) separates the sense wires resulting in a maximum drift distance of 0.55cm. The middle layer is staggered by a half cell relative to the other two layers to help resolve the left-right ambiguity.

The vertex chamber spans only 20° in azimuth. In the data here, there are on average only 2 to 4 tracks crossing it. Because of tracks from interactions in the beam pipe and from decays (the average number of such tracks is 30-40% of the number of z-chamber tracks), tracks often point to separate beam line vertices, resulting in ambiguities as to which is the true event vertex. The spectrometer and the TOF vertex information can be used to check the z-chamber vertex. The resolution in z of spectrometer tracks projected to the beam line is about 1mm. However, there are usually only 1 or 2 tracks in the spectrometer and they may not be primary tracks. In E735 data analysis, spectrometer tracks are accepted as primaries if their z vertex matches the TOF or z-chamber vertex. The TOF primary vertex resolution is about 4cm.

Using actual data rather than a Monte Carlo simulation to test the neural network introduces several real-world complications. Under ideal conditions the chamber achieved 100μm drift resolution, but due to the high background radiation levels in the C0 interaction hall, the z-chamber was normally run at a lower than ideal voltage to extend its lifetime. This resulted in 500μm drift resolution, which in turn gave 0.5cm vertex resolution for a single track. Also, the drift times refer to the distance of closest approach of the tracks to the wires rather than the distance from the point where the tracks cross the sense wire planes. Thus a given drift time will correspond to a different point in space depending on the angle where the track crosses the chamber. Due to the single field wire geometry of the z-chamber, ionization from a single charged track often reaches more than one sense wire in a given layer, especially for tracks at low angles relative to the beam line.

The drift chamber data here was obtained from events recorded on tape by experiment E735 at the Tevatron proton-antiproton Collider during a run in 1988-1989. Off-line the drift times have had pedestal values subtracted. Also, the times were converted to distances using a piecewise linear time to distance relationship made of 4 linear segments, each having a different slope for the different electron velocities across the drift cell. Because of the multiple hits per track problem, if two or more hits were contiguous in the same layer, only the hit with the smallest drift time was used by either the fitting methods or the neural network. (The multiple hit problem is not general to most other drift chambers so the filtering here of only the smallest drift time hits wouldn't have to be done when applying the neural network method to other chambers.) Despite the calibration and the filtering of multiple hits, the data include many "real world" problems, such as noise, inefficiencies and limited drift resolution, that present a rigorous test of the neural network method.

## 3. Z-Chamber Event Vertex by Track Fitting:

Here we give a brief description of the more conventional methods used in the E735 off-line analysis to determine the primary event vertex with the z-chamber . The z-chamber analysis is described in more detail in ref. [8]. A first approximation of the vertex (called the "global" vertex) is made using either the TOF vertex or a method based on ref. 9. In the latter method, a distribution in z along the beam line is calculated whose maximum occurs at the z of the event vertex. (This is similar to the z vertex distribution of the neural net mentioned above but derived in a different way.). This distribution is derived by first dividing the vertex range into 1cm bins. Beginning from one end of the vertex range, a "track" is found between the z bin and a hit in the outermost layer of the z chamber. The residuals between that track and all the hits in the other 2 layers are found (including both left-right possibilities for every hit). Similarly, the residuals are found for all other tracks between the z bin and the other hits in the outermost layer. The negative of the sum of the square of these residuals is entered into an exponential whose value is assigned to that z bin. The process is repeated for all other z bins. The maximum of the resulting distributions is shown [9] to occur at the event vertex. (There are ambiguities if there are peaks in the distribution which are only slightly smaller than the maximum peak.) Fitting the distribution of the differences between these global vertices and the vertices of projected spectrometer tracks gives a resolution of 1.5cm for the global vertex.

Individual tracks are found by defining 0.5cm wide "roads" between hits in the inner and outer layer and requiring that there be a middle layer hit in the road. Trying different fits for the different left-right possiblities of the drift distance from the wires, the fit with the lowest chi-square is chosen as the track.

Those z-chamber tracks which point within $\pm 10$cm of the global vertex are used to get a better vertex. If there is only one such track then its vertex is used as the event vertex. If there are 2 or more such tracks then they are re-fit requiring a common vertex. This gives a vertex resolution of 0.5cm. This final "track fit vertex", as shown in figure 2a, is used to train the network and to determine the network performance. Note that since we use actual data, we do not know the "true" primary event vertex. The neural network vertex here is always compared to the fit vertex.

## 4. Track Vertex Finding with a Neural Network:

To apply the neural network method to the z-chamber, it is divided into overlapping 18 wire sub-sections as shown in fig. 2b. The 18 wires correspond to 18 input units in a 3-layer feed forward network as show in fig. 3. Reference [5] describes similar track vertex finding in 15 wire z-chamber sections. The activation values of the input units are linearly proportional to the drift distance values of the corresponding wires. To avoid ambiguity between a wire with no hit and the case of a track passing at the wire and giving zero drift distance, a minimum value is used for hit wires. Here a wire with no hit has 0.0 activation in the corresponding input unit and hit wires have activation between 0.2 and 1.0

The 62 output units of the net correspond to 62 1.0cm bins along the beam line. As indicated in fig. 3, the position of the vertex will be indicated by the center of a Gaussian (of sigma 1.0cm) shaped cluster of 3 or 4 activated output units with all other output units off. Bins 2-61 correspond to a ±30cm range from the center of the given 18 wire section. Bin 1 is for vertices located less than -30cm from the center of the section and bin 62 for vertices greater than +30cm.

The middle layer of the network has 128 "hidden" units. This number was chosen somewhat arbitrarily. Neural net performance generally seems to improve with increasing numbers of hidden units and the maximum number of inputs in a currently available neural network chip is 128 (Intel "ETANN" chip, see section 6). A bias unit (a unit whose activation is fixed to 1.0) is connected to all hidden and output units. Fig. 3 shows that each hidden unit is connected to every input unit and the bias unit. Similarly, every output unit is connected to each hidden unit and the bias unit.

The activation of a hidden or output unit is proportional to the sum of the activations of units connected to it multiplied by a "weight" for each connection. For hidden unit j:

$$\mathrm{sum}_j = \sum_j w_{jk} a_k$$

where $a_k$ is the activation of input unit k, $w_{jk}$ is the weight between input unit k and hidden unit j, and the sum is over all input units. The activation of hidden unit j is a sigmoidal function (fig. 4) of this sum. Here the particular function used was:

$$f_j(\mathrm{sum}_j) = 1.0 / (1.0 + \exp(-\mathrm{sum}_j)) = a_j.$$

The sequence of actions to determine the vertex of tracks in a sub-section goes as follows (see also fig. 3). A track traverses an 18 wire sub-section creating ionization in three of the sense wire cells. The drift times of the electrons to the sense wires are converted to drift distances as describe in section 2. The three corresponding network input units will have activations between 0.2 to 1.0 proportional to these drift distances. All other input units will have 0.0 activations. The activations of the input units are passed to each of the hidden units, multiplied by a weight value that is unique to each input-hidden unit connection. Each of the hidden units then is activated according to the sigmoidal transfer function. The hidden unit activations are passed to each of the output units, again with each connection having a unique weight. The output units determine their activation with a linear function (fig.4). (Linear functions for the output units gave slightly better results than sigmoidal functions.) The network's track vertex corresponds to the z position of the output unit whose value is largest. Averaging over the cluster of bins around the maximum bin gives a somewhat more accurate vertex value. Fig. 5 shows some examples of tracks in 18 wire sub-sections..

Note that the network has no a priori information on which side of the wire the track passed. This left-right ambiguity must be resolved by the network using the the pattern of the hits and the fact that the middle layer is staggered by a half cell.

The values of the weights are the crucial factors which determine the response of the network to the inputs. Determining the weights is called the "training" of the network. Here the method used to train the network is the back-propagation technique described in ref. [10]. This is an iterative procedure in which multiple passes are made through a large "training set" of events, with the weights adjusted slightly after each event so as to reduce the overall error in the output for that event. Once a suitable set of weights is found, their values are fixed.

In back-propagation the weight between output unit i and hidden unit j is modified according to

$$\Delta w_{ij} = \eta (a_i - t_i) a_j f'_i(\mathrm{sum}_i)$$

where $t_i$ is the target value for output unit i, $\eta$ is a constant "learning" coefficient (0.5 used here) that determines the weight change step size, and $f'_j(\mathrm{sum}_j)$ is the derivative of the output function with respect to changes in $\mathrm{sum}_j$. If k is an input unit and j is a hidden unit, the formula becomes

$$\Delta w_{jk} = \eta a_k f'_j\left(\mathrm{sum}_j\right)\sum_i f'_i(\mathrm{sum}_i)(a_i - t_i)w_{ij}$$

where the sum is over all output units. The differences between the target values and the output values are "propagated" back to the weights between the hidden and input units. A modification of the procedure is to calculate the averages of $\Delta w_{ij}$ and $\Delta w_{jk}$ over several patterns and then make the changes.

For a set of such input-target patterns, the back-propagation performs a minimization of the sum:

$$E = \sum_p\left(\sum_i\left(a_i^p - t_i^p\right)^2\right)$$

where p refers to each pattern in the training set. After iterating through the training set until the weight changes become small, the weights are fixed and the net tested on an independent set of patterns. Further training can continue if the performance on the independent set is not deemed sufficient.

The training set here for the 18 wire sub-section network was created from z-chamber events with tracks and vertices found by the methods described in section 3. For a given event, 18 wire sections were examined, starting from one end of the chamber and stepping through the chamber 1 wire at a time until 3 fit track hits were contained within the section. Fig. 5 shows examples of sub-section patterns. For such an event, the 18 drift distances of the wires and 64 target values for the output units (the fit track vertex position given as a Gaussian of sigma 1.0 as shown in fig.3) were recorded. The training set consisted of 5700 patterns with 1 fit track only (i.e. 3 hits only), 2850 patterns with 1 fit track plus at least one background hit not used by the fitting program, 2850 patterns with 2 fit tracks (6 hits only) from the same vertex, and 600 patterns with all inputs 0.0 and all outputs 0.0. (The latter was to insure that the bias unit didn't produce a non-zero output when all inputs were zero). The training set did not include cases where there were tracks in the sub-section from different vertices or where there were multiple contiguous hits per layer for a single track.

In general, the larger the training set, the better the network performs on independent sets. This is intuitively reasonable since basically the network is doing a mapping from "drift distance space" to "track vertex" space. The more points sampled in these spaces, the better the network learns the mapping. Even with these 12000 patterns in the training set, the training is not perfect and the output response to a track of a given angle will vary somewhat depending on where it goes through the sub-section.

After roughly 4 million iterations through the training set, the network's performance was tested on an independent set of events with same cuts (i.e. require tracks to come from same vertex and no multiple contiguous hits). Fig. 6 shows distributions of the difference in the network's vertex and the target vertex from the fit tracks. The net vertex here is the average over the range of ±2 bins around the bin with the maximum activation. The track vertex resolution (resolutions here are expressed as the sigma of the Gaussian fits) for single track events with no noise is 0.72cm. If there are background hits the resolution is 1.7cm and for 2 track events the resolution is 2.1cm. In all three cases there are non-Gaussian tails, especially for the last two cases. These tails arise primarily because of track ambiguities. Fig. 7 shows cases of ambiguous tracks. Even with only three hits, there can be ambiguities, especially for low angle tracks. Increasing the number of hits because of backgrounds or extra tracks further increases the possibility of ambiguities. Note that the network often gives some output response at each of the different vertex possibilities.

## 5. Primary Event Vertex:

For a whole proton-antiproton event, the primary vertex is determined by feeding the drift times of overlapping 18 wire sections (fig. 2b) sequentially into the network simulation and summing the outputs as described earlier. If this was done with neural network hardware, the process would be entirely parallel. Ideally, the network would be trained to handle correctly any possible input pattern. However, to reduce the range of inputs and the training set size, it was required here that before a network output was added to the overall vertex distribution, each layer of the given sub-section had to have between 1 and 3 hits and total number of hits less than 7. This meant that sub-sections with only one or two background hits or a large cluster of hits were ignored. Since the net was not trained on such patterns, the network output vertices for these patterns were unreliable. However, even with these cuts, all events in the independent test set had at least one sub-section that passed the cuts. For a trigger, these simple cuts could be calculated in parallel with the network hardware by using fast logic circuits to decide whether the net's output should be passed onto the next level that adds all of the networks together.

The outputs of the nets are added as shown in fig.2b. The maximum of the resulting distribution indicates the position of the primary vertex (fig.2a). Fig.8 and 9 show several events. Fig.10a. shows the distribution of differences between the fit tracks vertex and the net vertex.The sigma of the Gaussian is 1.4cm. There are long non-Gaussian tails primarily due to cases where there are different tracks pointing to different vertices (the network chooses a different vertex than the one the fit chose) and where clusters of hits cause the network to give an erroneous vertex (fig.9). The network maximum peak is found to be within ±10cm of the fit vertex in 87% of the events. This compares to about 80% of the TOF vertices found within 10cm of the fit vertex (fig. 10c). If no cuts are made in the test set on the number of hits per layer and on the total number of hits in the sub-sections, the sigma is 1.5cm and 82% of the entries are within ±10cm of the fit vertex.

Fig. 10b shows a distribution of differences between the fit vertex and the network distribution peak nearest to the fit vertex. (Of course, which peak is the nearest isn't known before fitting). Here the sigma is 1.3cm and the tails are reduced. There is a peak within 10cm of the fit vertex in about 98% of the events. This indicates that the net almost always finds at least one track pointing towards the fit vertex but that the sum of the outputs is not always largest there. This can be due to genuine primary vertex ambiguities. For example, note in fig. 9d that the TOF vertex agrees with the network vertex and not the fit vertex. The failure of the largest peak not to be near the fit vertex can also be due to cases where tracks pointing towards the fit vertex do not give large enough output activations (the network output decreases for cases of track ambiguities especially). In addition, the network track vertex resolution may not be good enough for the vertices of tracks in two separate sub-sections always to overlay well enough to make one large peak.

The z-chamber spans a small section of the total solid angle and so the number of tracks in it is small. Since the percentage of tracks from secondaries will fluctuate from event to event, the vertex ambiguities can arise regardless of the method used. From scanning events by eye and comparing the fit vertex to the TOF and spectrometer track vertex, the fit event vertex is ambiguous in about 5%-10% of the events. For a chamber that covered a larger portion of the solid angle (e.g. if there were 3 other z-chambers surrounding the beam pipe in a box arrangement), the performance of the network method here would improve since the number of tracks pointing at the true primary vertex would increase.

For the results shown here, the sub-sections were stepped every 2 wires through the chamber. In the overlapping regions of the sub-sections, tracks nearly normal to the beam pipe can produce outputs in more than one sub-section network. This multiple counting is not really a problem (other step sizes were tried and results varied little) since it just adds some redundancy to the vertex finding, which helps to smooth out the non-uniformity in network response mention earlier. If it is desired to eliminate multiple counting the net could be trained, for example, to ignore tracks totally contained within the overlapping section of the net on, say, the left hand side.

## 6. Discussion:

We've shown that a neural network can be trained on actual experimental data to find the vertex of tracks in a sub-section of a drift chamber. Adding the outputs of such networks from over-lapping sub-sections spanning the entire chamber provides the primary vertex of proton-antiproton events. The performance is nearly as good as a more conventional off-line fitting algorithm used at present in E-735 analysis and is better than an off-line TOF vertex. To achieve this, the net had to be trained with data which had many complicating factors such as 500$\mu$m drift resolution, noise hits, and secondary tracks. For a single 18 wire sub-section the track vertex resolution was 0.72cm for single tracks and 2.1cm for 2 tracks. Adding all sub-section outputs together gave a primary vertex resolution of 1.4cm. Because of track and vertex ambiguities, there are long non-Gaussian tails to both distributions. The performance of such a method if applied to other detectors should improve as the sample of tracks increases with larger solid angle coverage. Better detector resolution, more layers, and the absence of multiple hits for single tracks would help, as well. Improving the network performance is also being studied, such as varying the number of hidden units. The performance here, though, may already be adequate for some trigger applications.

For on-line use the neural network method can be compared to the TOF vertex, which might conceivably be available on-line. The TOF vertex used here came after timing and pulse height calibrations, corrections for time slewing (variation of timing with pulse height) and for multiple hits in individual counters [6]. The TOF vertex resolution is about 4cm as compared to 1.4cm for the neural net. The neural net vertex resolution would improve substantially with better drift resolution and more drift chamber layers (to reduce the ambiguities), whereas it would be very difficult to improve on the 250psec timing resolution of the E735 TOF system. Also, the neural net method provides for multiple vertices in a straightforward way.

The approach here of training nets on sub-sections might be applied to other tasks. For example, instead of finding the intercept of tracks with the beam line, the intercept of tracks with a detector at some radius from the beam line could be determined. Correlation of tracks with calorimeter showers or muon counter hits could then be done.

A hardware implementation of a neural network vertex finding system may soon be feasible using new VLSI chips. For example, the Intel ETANN chip [11] has 64 totally interconnected neurons and can be configured for 128 inputs and for multi-layer use. The neural network here would need 3 such chips: 2 for the hidden layer and 1 for the output layer. Circuit simulations and hardware tests will be needed to establish the suitability of these chips to the vertex finding type application discussed here. It's encouraging, though, that the number of neurons in the first generation of chips is already of the scale required.

**Acknowledgements:**

We thank Theo Alexopoulos and the Wisconsin z-chamber group for z-chamber data and information. Thanks also to Chiho Wang and Eliane Lessner for programming assistance. Thanks to E. W. Anderson for his support. This research supported by Fermi National Accelerator Lab.

* Fermilab is operated by the Universities Research Association under contract with the Dept. of Energy

**References:**

1. K. Fukushima, "Neural Networks", vol. 1, num. 2 (1988)119.
2. R. Gorman, T. Sejnowski, "Neural Networks", vol. 1, num. 2 (1988)75-89.
3. B. Denby, "Computer Physics Communications", 49(1988) 429-448.
   C. Peterson, NIM A279(1989) 537-545.
4. D. Cutts et al., "The Use of Neural Networks in the D0 Data Acquistion System", presented at conference Real-Time '89, Williamsburg, Va. (May 1989), to be published.
5. B. Denby, E. Lessner, C.S.Lindsey, 'Tests of Track Segment and Vertex Finding with Neural Networks", proceedings of "1990 Conference on Computing in High Energy Physics", pp.211-218, Sante Fe, N.M., April 1990.
6. Bannerjee et al., NIM A269 (1988)1211.
7. E.W.Anderson et al, "A Scintillator Hodoscope at the Tevatron Collider", accepted NIM A.
8. T. Alexopoulos et al, "A One Meter Long Low Mass Mini-Drift Chamber Used at the Tevatron Collider", to be submitted to NIM A.
9. Y. A. Yatsunenko, "Vertex Reconstruction without Track Reconstruction", NIM A287(1990)422.
10. D. Rumelhart et al, "Parrallel Distributed Processing, Explorations in the Microstructure of Cognition", vol. 1, ch. 8, MIT Press, Cambridge, Mass.
11. M. Holler et al., "An Electronically Trainable Analog Neural Network (ETANN) with 10240 'Floating Gate' Synapses", proc. Int. Joint Conf. on Neural Networks, vol II, pp.191-196, Washington D.C., June 1989, IEEE Catalog 89CH2756-

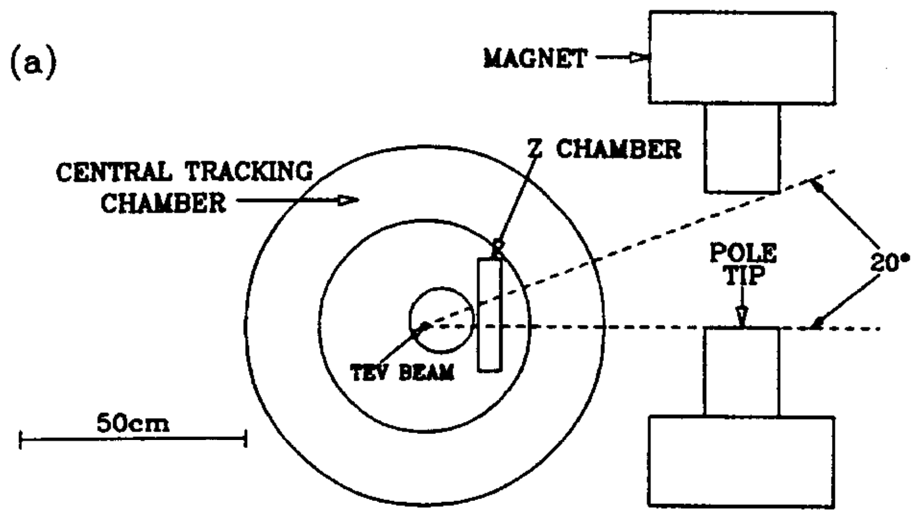

Fig.1a. Cross-section of E735 detectors at the Tevatron C0 intersection hall. Spectrometer detectors not shown. (Note that the Tev beam is not centered in beam pipe at C0.)

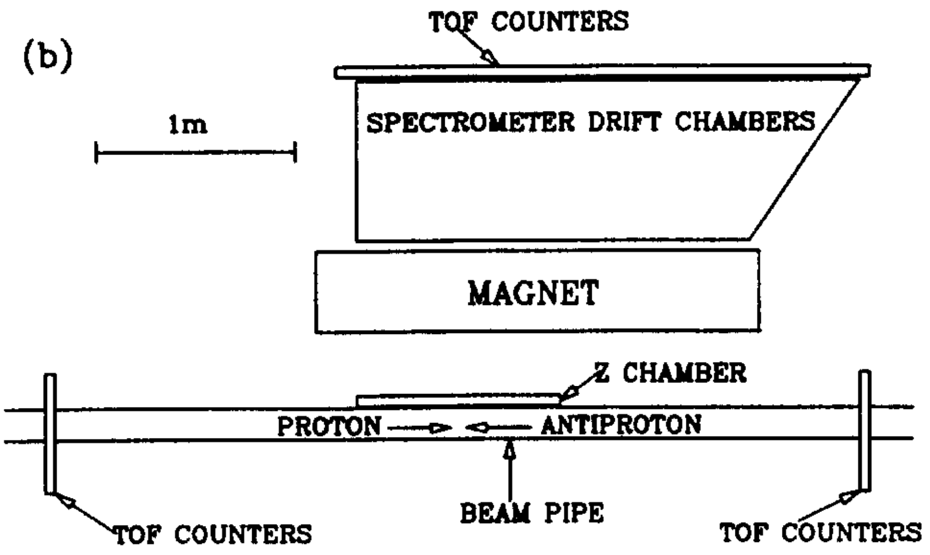

Fig. 1b. Plan view of E735 detectors. For clarity, central tracking chamber and other detector in region surrounding beam pipe are not shown.

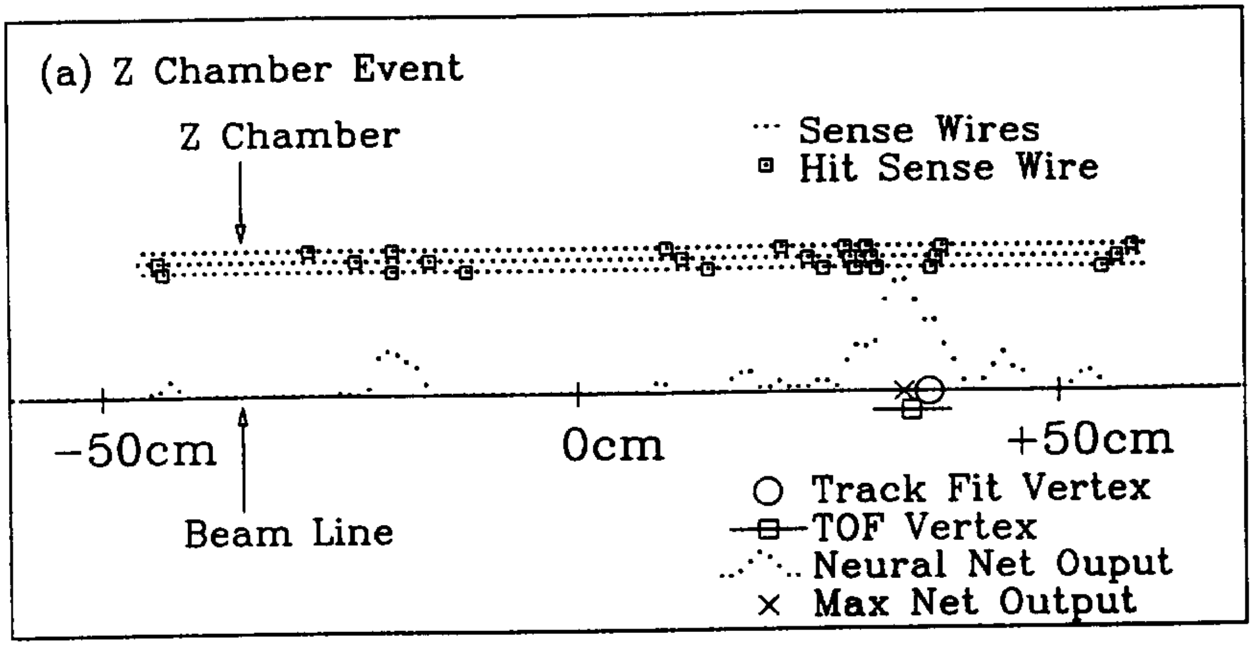

Fig.2a. A proton-antiproton event showing hits in z-chamber. The beam pipe, z-chamber enclosure and field wires are not shown. Shown are the primary vertex found by track fitting, the time-of-flight (TOF) vertex (displaced below beam line for clarity) and the vertex given as the maximum in a neural network output distribution. Each point in the network distribution represents a 1cm bin along the beam line.

(b) Schematic of Z Chamber Neural Network System

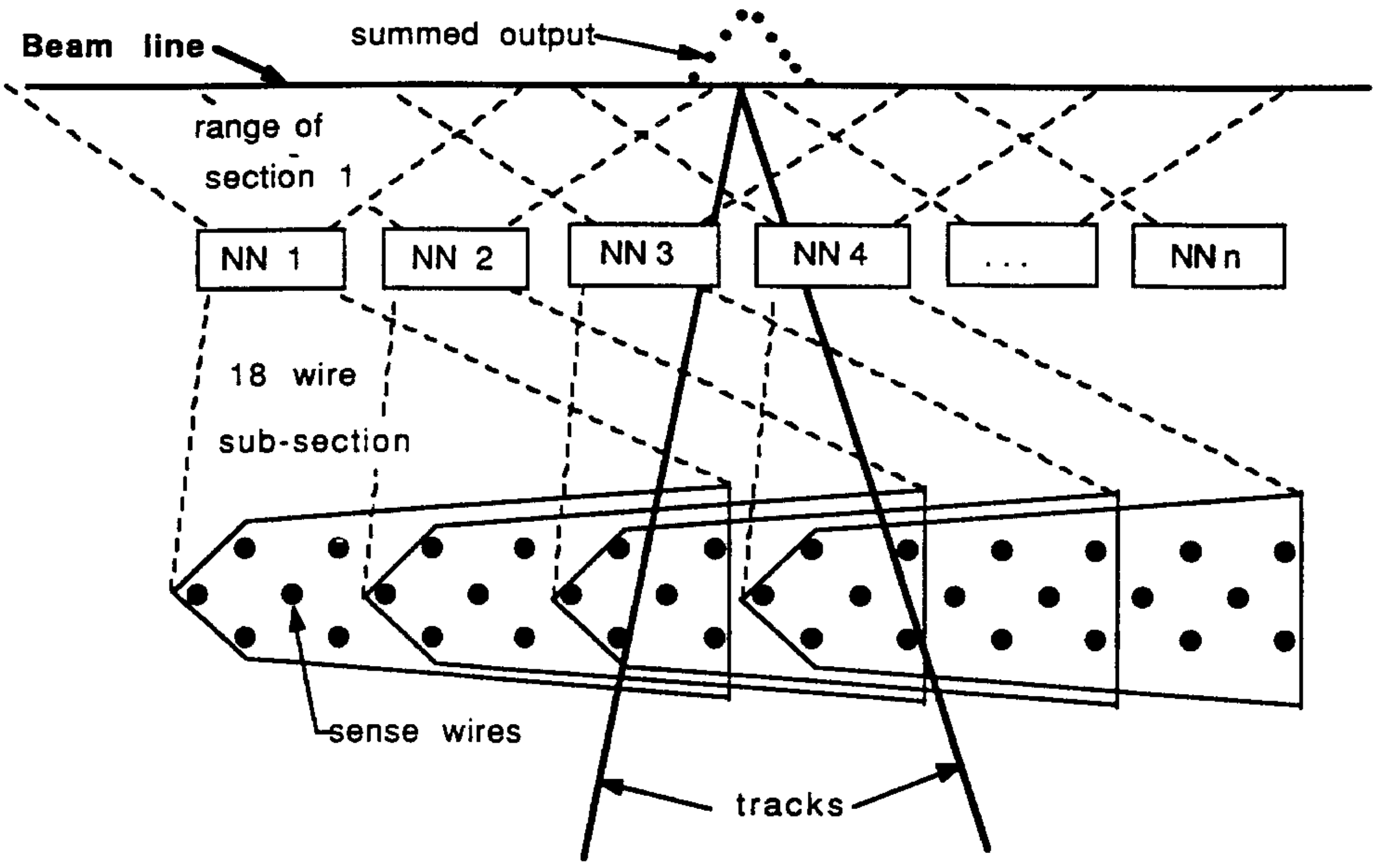

Fig.2b. Signals from overlapping 18 wire sub-sections of the z-chamber are fed into neural networks. The output of each network represents the possible position of the intercepts of tracks in the sub-section over a portion of the beam line. Adding the overlapping network outputs for two tracks from an event vertex gives the distribution shown.

NEURAL NETWORK FOR Z-CHAMBER SUB-SECTION VERTEX

Input = 18 Sense Wire Drift Times
Output = 60 1.0cm Bins From −30cm to +30cm
+ 1 Bin for Z<−30cm + 1 Bin for Z>+30cm

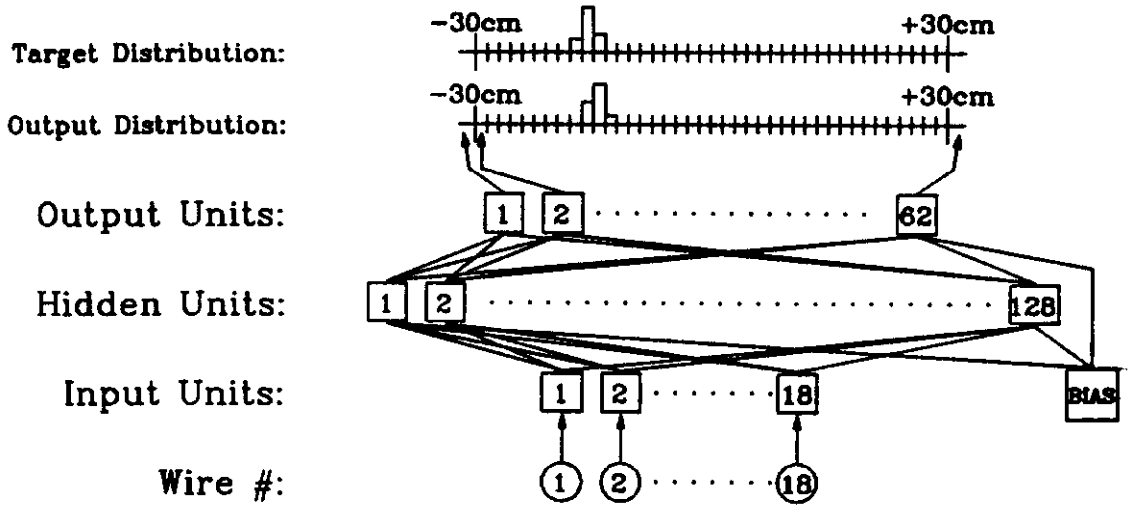

Fig. 3. The neural network architecture used to determine the vertex of tracks in 18-wire z-chamber sub-sections. All input units and the bias unit are connected to all hidden units. All hidden units and the bias unit are connected to all output units. Only a few of the connections are shown. The bias unit has an output activation fixed to 1.0.

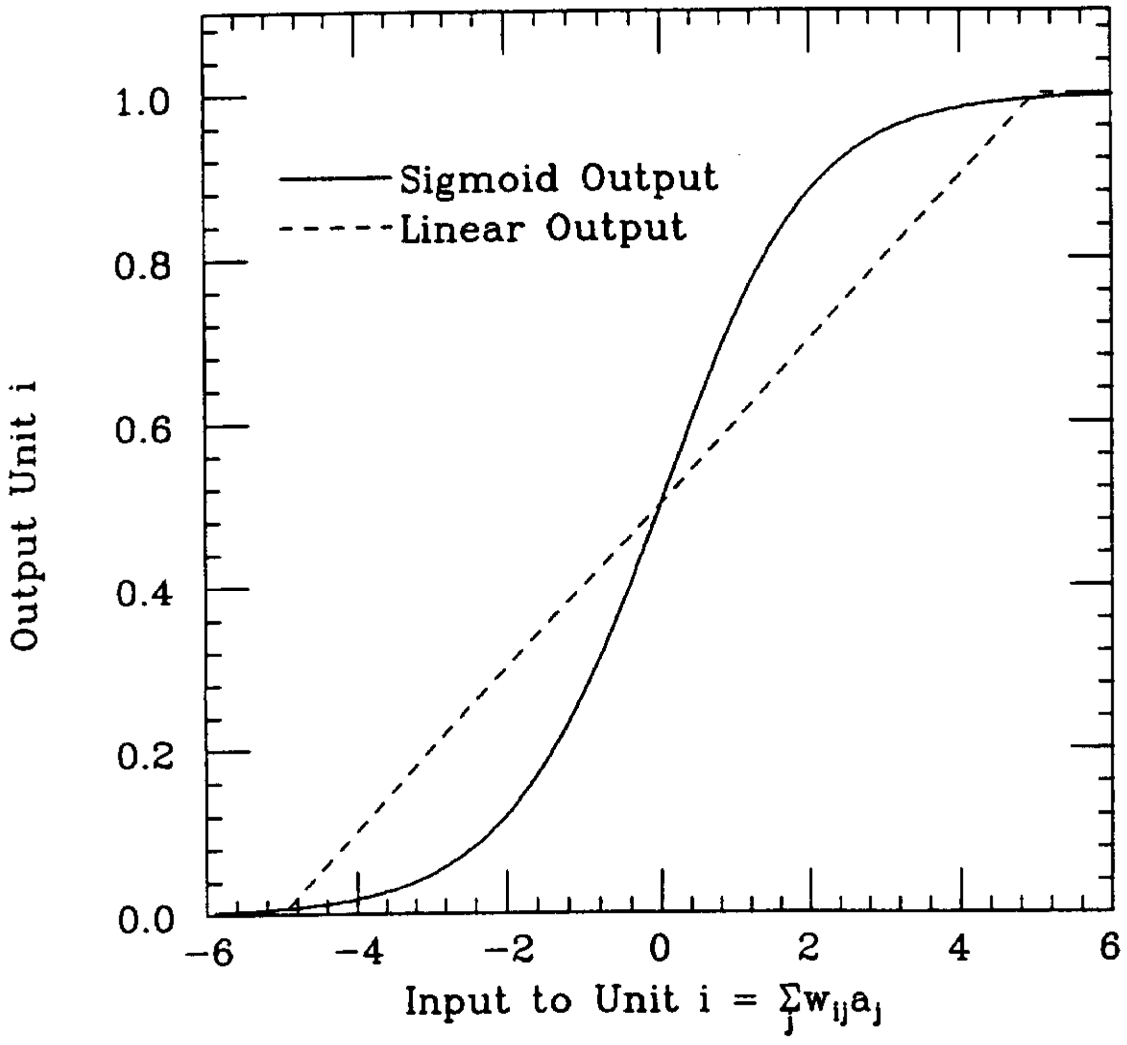

Fig.4. Neuron output activation as function of the weighted inputs. The sigmoidal output was used for the hidden units, linear for the output units.

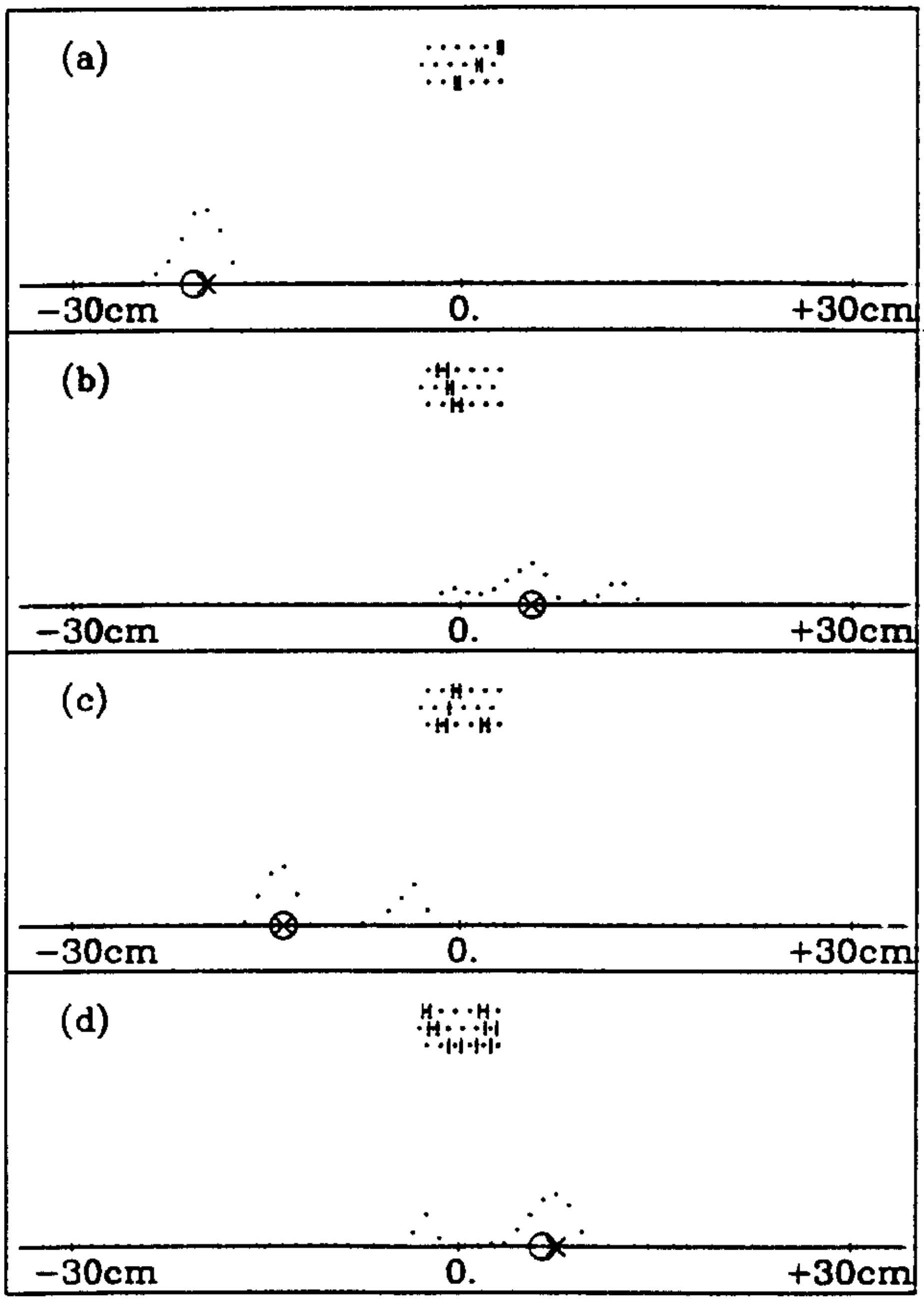

Fig.5. Examples of hits in 18 wire sub-sections of the z-chamber. The fit vertex is compared to the neural network vertex given as the position of the maximum output unit.
(a)-(b) single tracks, (c) single track plus background hit, (d) 2 tracks.
Vertical lines represent drift distances from the sense wires with left-right ambiquities.
Otherwise, symbols are same as in figure 2a.

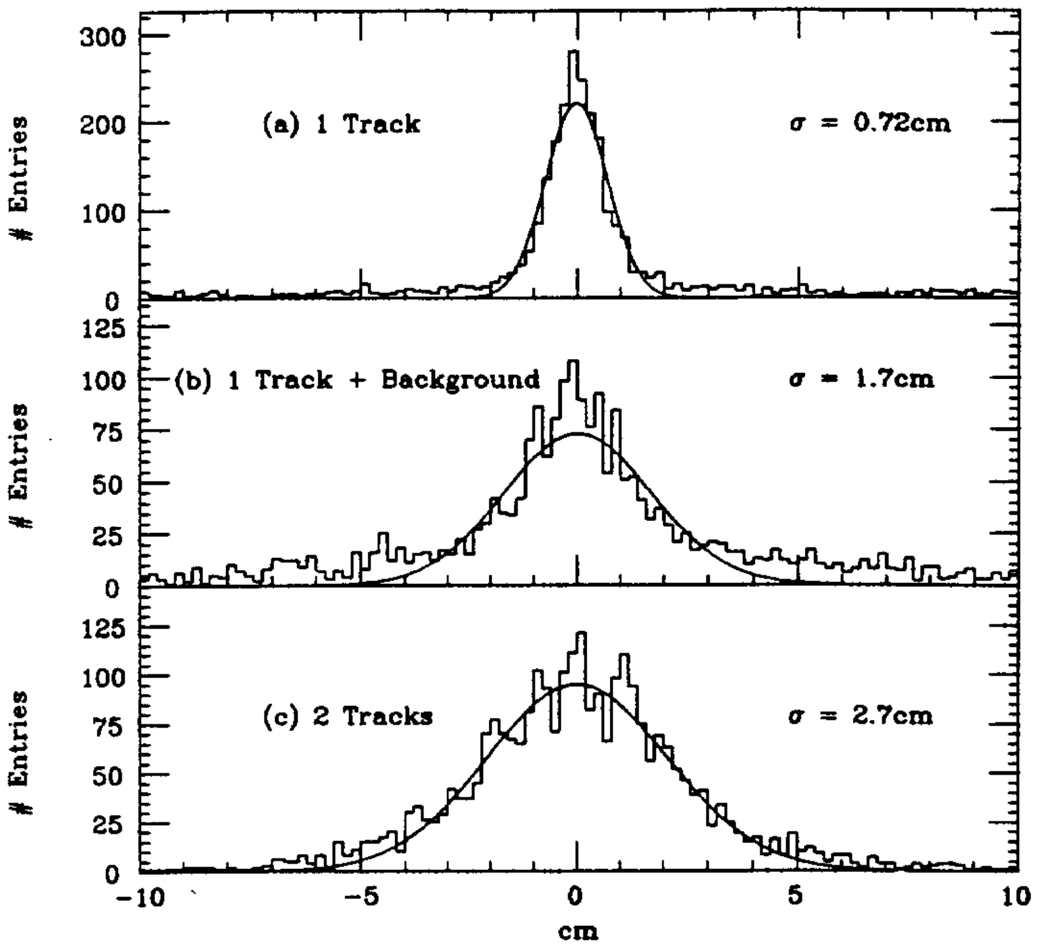

Fig.6. Distributions of differences between network vertex and fit vertex for 18-wire sub-section events. (a) single tracks in sub-section, (b) single tracks plus background hitrs, (c) 2 tracks.

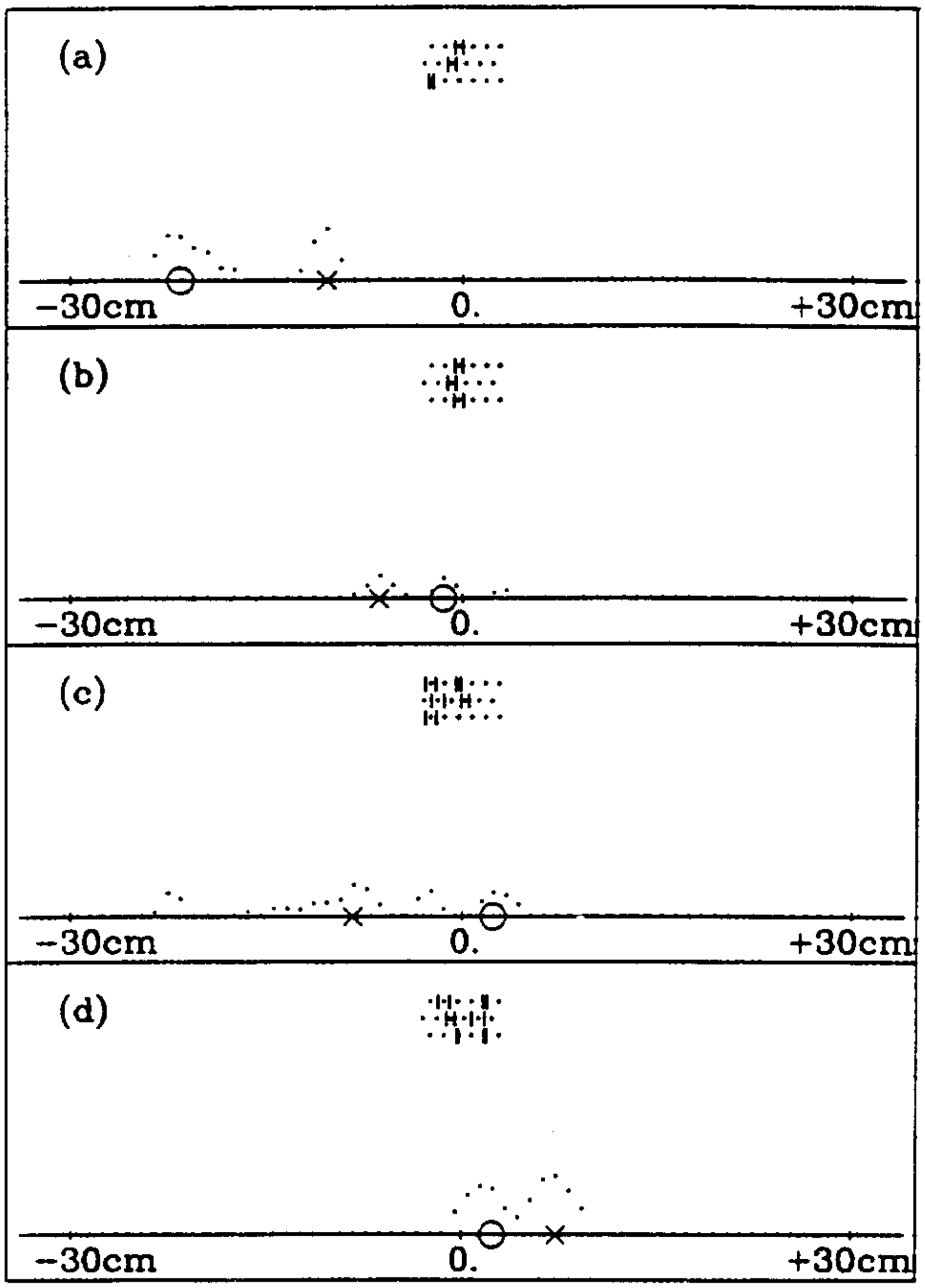

Fig.7. Examples of 18 wire sub-section events in which the network vertex and fit vertex disagree. (a)-(b) single tracks, (c) single track plus background hit, (d) 2 tracks. Vertical lines represent drift distances from the sense wires with left-right ambiquities. Otherwise, symbols same as figure 2a.

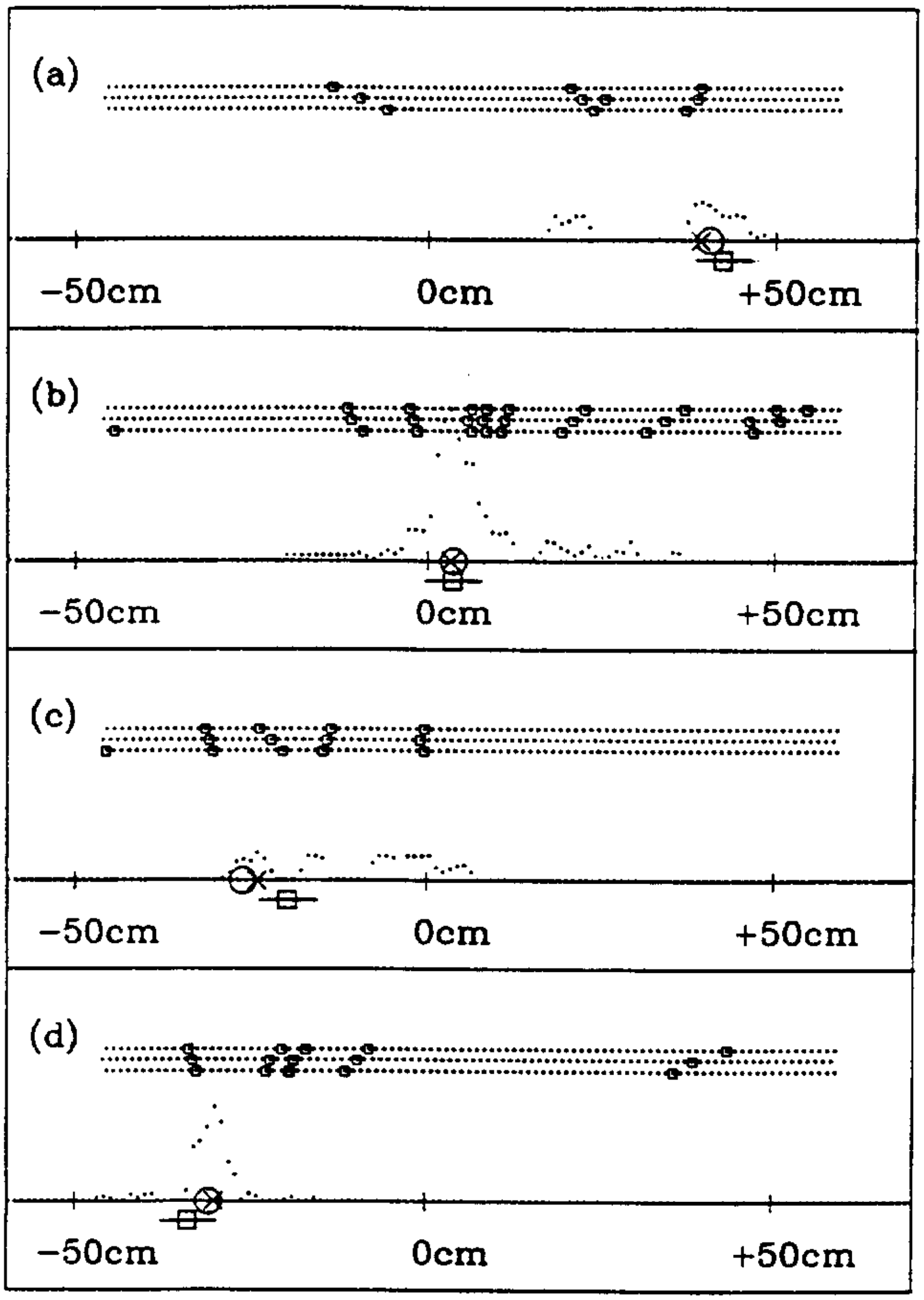

Fig. 8. Examples of whole z-chamber events in which all 18-wire sub-section networks have been added together. Symbols defined in fig.2a.

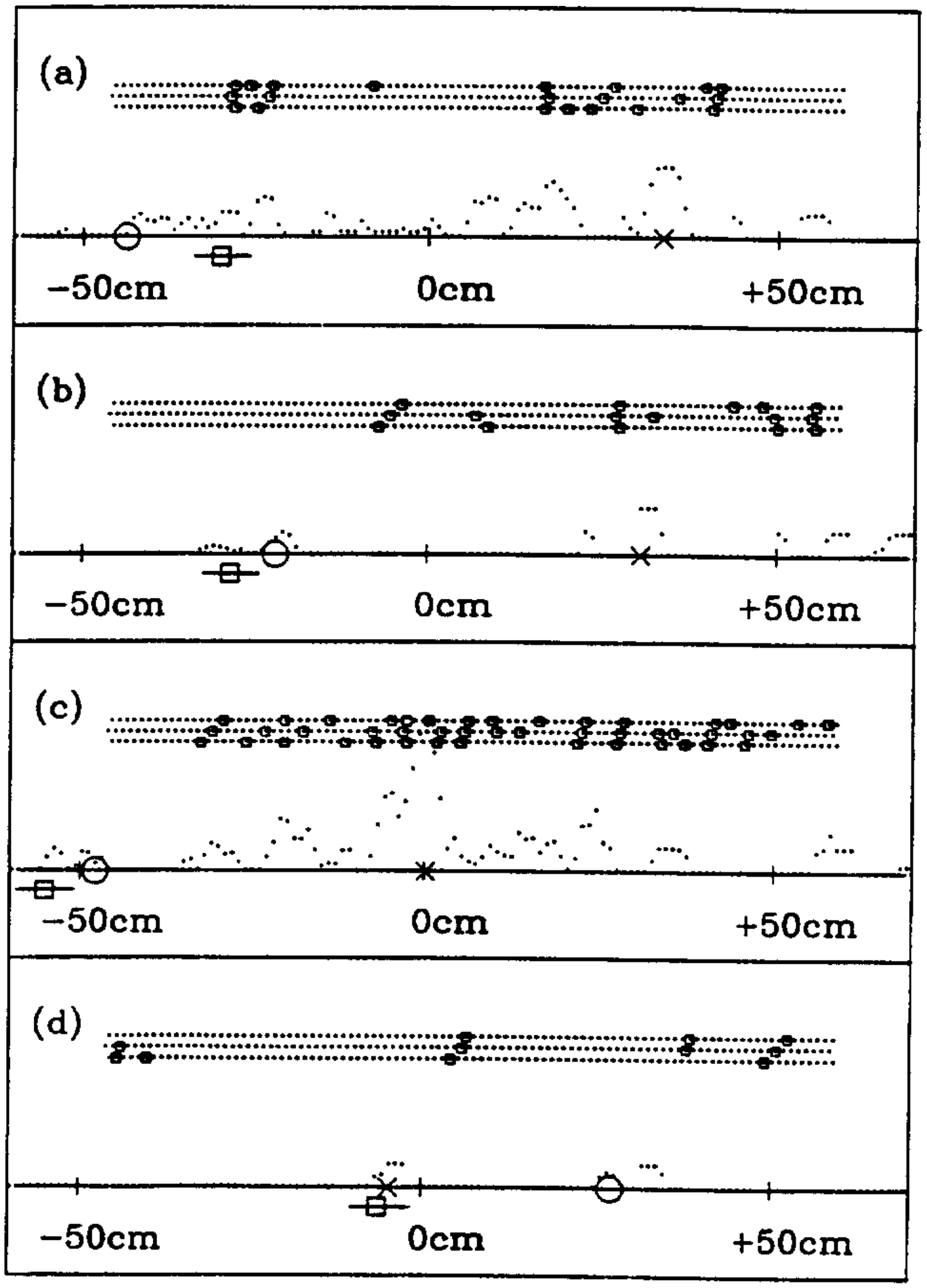

Fig. 9. Examples of whole z-chamber events in which the fit track vertex and network vertex disagree. Symbols defined in fig.2a.

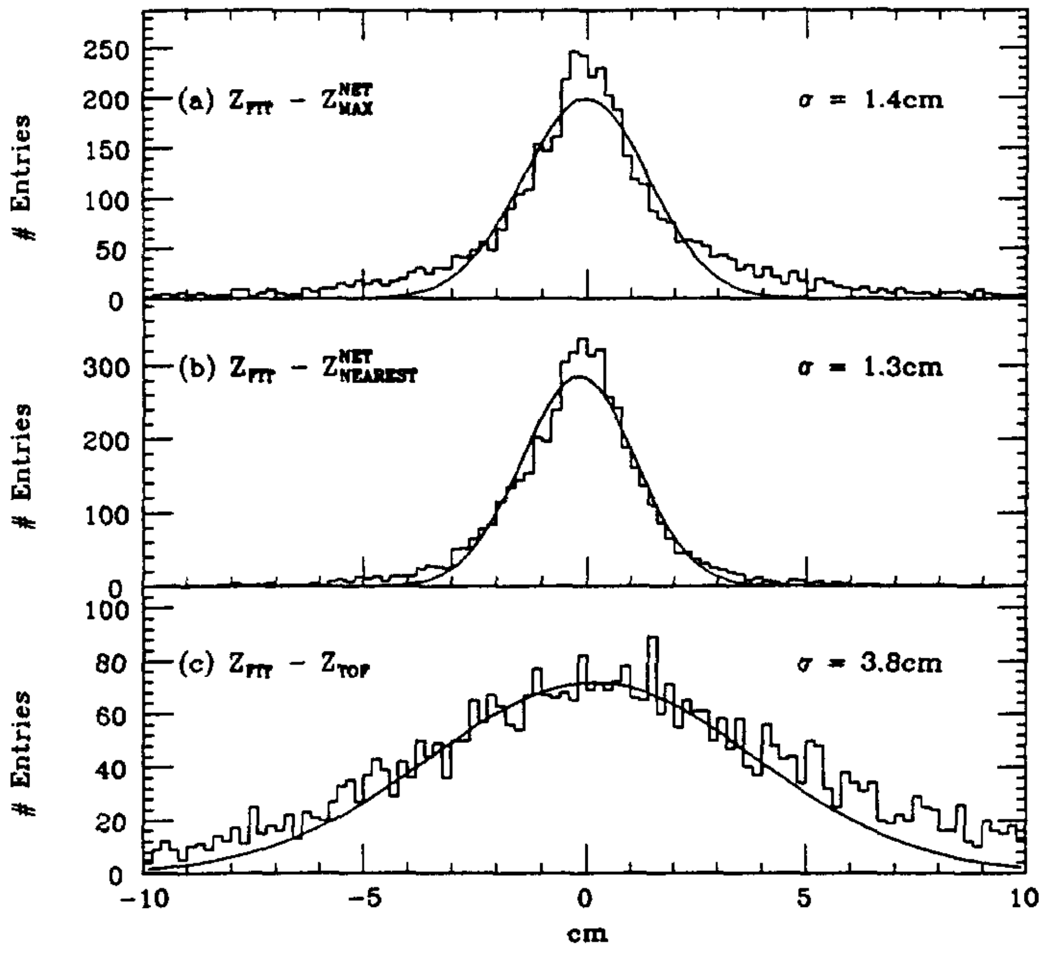

Fig.10. Distributions of differences between the fit vertex and (a) the position of the maximum in the network output distribution, (b) the position of the peak closest to the fit vertex, and (c) the TOF vertex.